FPGA 를 활용한 8bit Quantized CNN 가속기 시스템 제안

박건하, 신승제, 조연호, 조정훈*
경북대학교, *경북대학교

qkrrjsgk79@knu.ac.kr, soy3563@knu.ac.kr, bryan36@knu.ac.kr, *jcho@knu.ac.kr

A Proposal for an 8-bit Quantized CNN Accelerator System Architecture Using FPGA

Geonha Park, Seungjae Shin, Yeonho Cho, Jeonghun Cho*
Kyungpook National Univ, *Kyungpook National Univ.

요 약

본 논문은 소형 임베디드 장치의 추론연산 가속을 위해 FPGA 를 활용한 8bit Quantized CNN 가속기와 이를 포함하는 개발환경 및 리눅스 시스템에 대해 제안한다. 테스트 환경의 개발을 위해 MNIST 데이터셋을 예측하는 CNN 모델의 구현 및 NNoM(Neural Network on MCU)기반 8bit 양자화를 적용했으며, 가속기 IP 설계 및 합성 후 이에 대한 테스트 결과를 제시했다.

Ⅰ. 서 론

인공지능 연산에 널리 사용되는 GPU 시스템은 병렬연산에 최적화되어 있어 인공지능 모델 구조에 독립적으로 연산 가속이 가능하다. 이러한 범용 시스템보다 향상된 성능을 얻기 위한 근래의 인공지능 가속기 연구는 딥러닝 모델의 특정 레이어 단위 연산을 가속하는 IP 를 설계하거나, 모델 전체 구조를 IP 로 설계하여 성능을 높이는 방법이 시도되고 있다.

특수목적으로 사용하는 소형 임베디드 장치에서는 모델 전체를 IP 로 구현하는 것이 유리하다. 이는 장치의 제한된 리소스로 인해 하나의 인공지능 모델만 사용하는 경우가 많기 때문이다. ASIC 형태의 가속기를 설계하는 경우 (단일 칩셋 형태) 타깃으로 하는 모델 구조의 변경에 대응하기 어렵다는 단점이 있다. 반면 IP 구조의 변경이 가능한 FPGA 활용 시, 설계와 검증에 따른 구조 변경에 유연한 대응이 가능하다.

모델 추론을 목표로 하는 인공지능 가속기의 경우 주로 모델 양자화, 가지치기 기법 등을 통해 모델을 경량화 한다. 이때 추론 정확도와 연산 소요시간의 trade-off 관계를 고려하여 연산 소요시간을 최적화하는 것을 모델 경량화의 최종 목표라고 볼 수 있다.

따라서, 본 논문에서는 CNN 모델의 8bit 양자화와 해당 모델의 추론 연산 가속을 위한 FPGA 기반의 8bit HW 가속기(DPU), 그리고 이를 지원하는 리눅스 시스템 구성을 제안한다.

논문의 구성은 다음과 같다. 2 장에서는 CNN 모델 및 양자화 결과, 그리고 CNN 가속기의 IP 합성 결과를 제시한다. 3, 4 장에서는 OS 와 IP 를 포함하는 전체 시스템을 제시하고 결론 및 향후 연구 방향을 소개한다.

Ⅱ. CNN 모델 구조 및 양자화 결과

사용한 CNN 모델은 다음 **표 1** 과 같다. FPGA 보드의 메모리 크기와 호환 여부를 고려하여 간단한 모델로 구성했고, 모델 테스트에는 MNIST 데이터셋을 사용했다.

표 1. CNN 모델 구조

layer	output	activation	info
Input	28x28x1	-	-
Conv1	12x12x16	ReLU	5x5, 16, stride 2
Conv2	5x5x32	ReLU	3x3, 32, stride 2
Flatten	800	-	-
FC-32	32	ReLU	-
FC-16	16	ReLU	-
FC-10	10	Softmax	-

8bit 양자화는 NNoM(Neural Network on MCU) 프레임워크를 기반으로 진행했으며, min-max 방법을 채택했다. 이는 각 레이어 단의 가중치 당 최대, 최소 값을 확인하여, 정수부와 소수점 부분의 표현 가능 비트 수와 함께 int8 범위로 표현하는 방식이다. 양자화 전/후 모델 추론 결과는 다음 **표 2** 와 같다. (추론 결과를 5 회 평균하여 계산, 5000 개의 MNIST 데이터 랜덤선택).

표 2. 양자화 전/후 모델 추론 및 리소스 비교

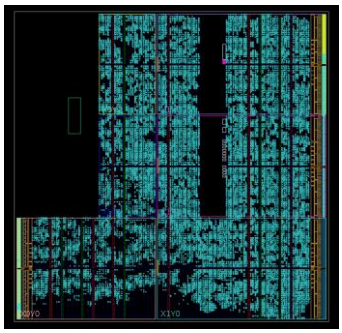
	Tensorflow2	NNoM
정확도	94.26 %	93.46 %-
가중치 파일 크기	427.4 KB	142.1 KB

위 모델 구조를 기반으로 설계한 DPU(Deep-Learning Processor Unit) IP 의 합성(post-synthesis) 결과 리소스 사용량은 다음 **표 3** 과 같다.

표 3. 가속기 IP 합성 결과 리소스 사용량 (Zybo Z7-20)

Resource	Estimation	Available	Utilization %
LUT	27462	53200	51.62
FF	2	17400	0.01
DSP	3056	106400	2.87
IO	84	125	67.20
BUFG	1	32	3.13

그림 1. 가속기 IP 합성 결과 사진



Ⅲ. 가속기 사용 전/후 결과(예상) 및 제안하는 시스템

가속기 적용 후 예상 소요시간은 다음 **표 4** 와 같다.

표 4. 가속기 IP 적용 전/후 추론 속도 비교

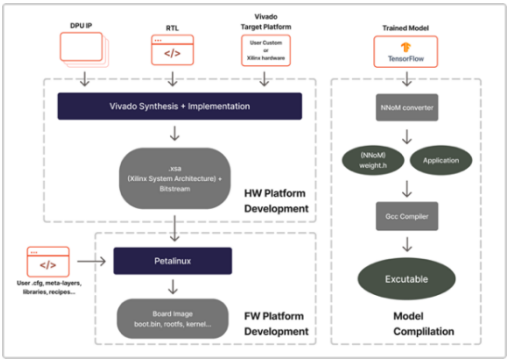
	적용 전	적용 후(예상)*
연산 소요 시간	69.097 sec	7.5m sec**

* 데이터의 테스트벤치 기준 입력 시점 0.105us, 출력 valid 신호 검출 시점 7.495us 이다. 검증 시스템 clock 은 10ns 으로, IO 전송속도와 clock skew 등을 고려하면 실제 연산 소요 시간은 이보다 1000 배 느린 7.5ms 일 것으로 예상된다.

** $T^{consume}=(Data^{in}-Data^{out})*delay^{IO\_access}*delay^{clk}=(7.495-0.105) * 100 *10 = 7.5ms$

가속기 IP, OS, 모델 및 응용프로그램을 포함한 전체 시스템은 **그림 2** 와 같다. HW 부분은 Vivado tool 을 기반으로 IP 를 설계 후, 바이너리 형태로 추출하여 보드 BSP 와 함께 커널 이미지를 생성한다. (Petalinux Tool 활용) 인공지능 모델 및 어플리케이션 부분은 TensorFlow 에서 학습된 모델을 NNoM 통해 경량화 하고, GCC 컴파일러를 사용하여 응용 프로그램을 빌드한다. 어플리케이션 영역에서는 디바이스 드라이버를 통해 PL 영역의 DPU IP 에 접근이 가능하다. 이를 위해서 PL 내부의 레지스터 IP 설정이 필요하다.

그림 2. 전체 시스템 구성



Ⅳ. 결론 및 향후 연구계획

본 연구에서는 1) Xillinx 사의 FPGA 을 활용하여 HW 가속기 IP 를 설계하고 이를 인공지능 어플리케이션과 함께 사용하는 리눅스 시스템에 대해 제안했다. 또한, 2) 8bit 로 양자화 한 CNN 모델의 DPU 사용 전/후(예상) 성능 차이를 비교했다.

이를 통해 타 환경에서 작성한 모델 어플리케이션 역시 해당 빌드 방식으로 적용할 수 있으며, 이를 FPGA 기반의 가속기로 소요 시간 단축이 가능하다는 결론에 이를 수 있었다. 이러한 소형 가속기는 전력 소모량 관점에서 GPU 에 비해 소형 임베디드 장치에서 이점이 있을 것이라 예상된다. 향후에는 이를 기반으로 정확도, 소요시간 그리고 전력소모량 관점까지 고려하여 이를 향상시킬 수 있는 새로운 IP 구조에 대해 제안하고자 한다.

ACKNOWLEDGMENT

본 논문은 2023 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 1711160343, 차량 ECU 응용소프트웨어 개발 및 검증자동화를 위한 가상 ECU 기반 차량레벨 통합 시뮬레이션 기술개발).

참 고 문 헌

[1] Raffaele Meloni," *PS-PL communication management*," May. 2022, (https://mdc-suite.github.io/miscellaneous/ps-pl-communication).

[2] Xilinx," *PetaLinux Tools Documentation: Reference Guide (UG1144)*, Oct 18, 2023, (https://docs.xilinx.com/r/en-US/ug1144-petalinux-tools-reference-guide/Introduction).